PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

FACULTAD DE LETRAS

Curso de procesamiento del lenguaje natural

César Antonio Aguilar

Facultad de Lenguas y Letras

13/06/2012

Cesar.Aguilar72@gmail.com

Semántica y PLN

Uno de los temas que ha cobrado gran actualidad en PLN es el análisis semántico .

No es un tema fácil, dada la complejidad que muestra explicar la semántica de una lengua natural, tanto para computólogos como para los mismos lingüistas.

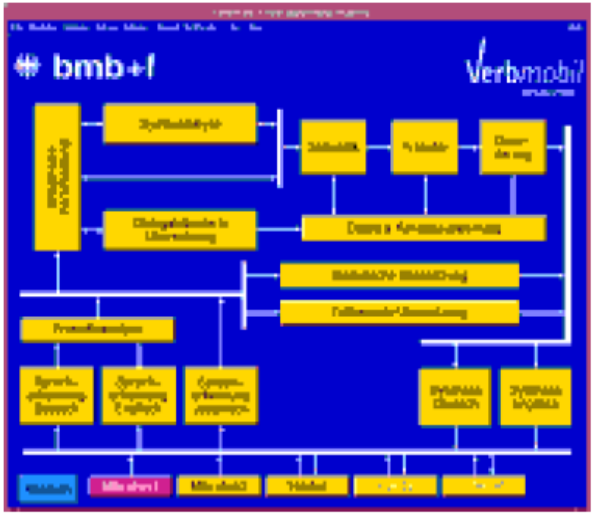
Como sabemos, el hecho de que la semántica sea difícil viene desde sus orígenes.

Platón y Aristóteles, según Rafael

¿Por qué queremos usar computadoras en cuestiones de semántica? (1)

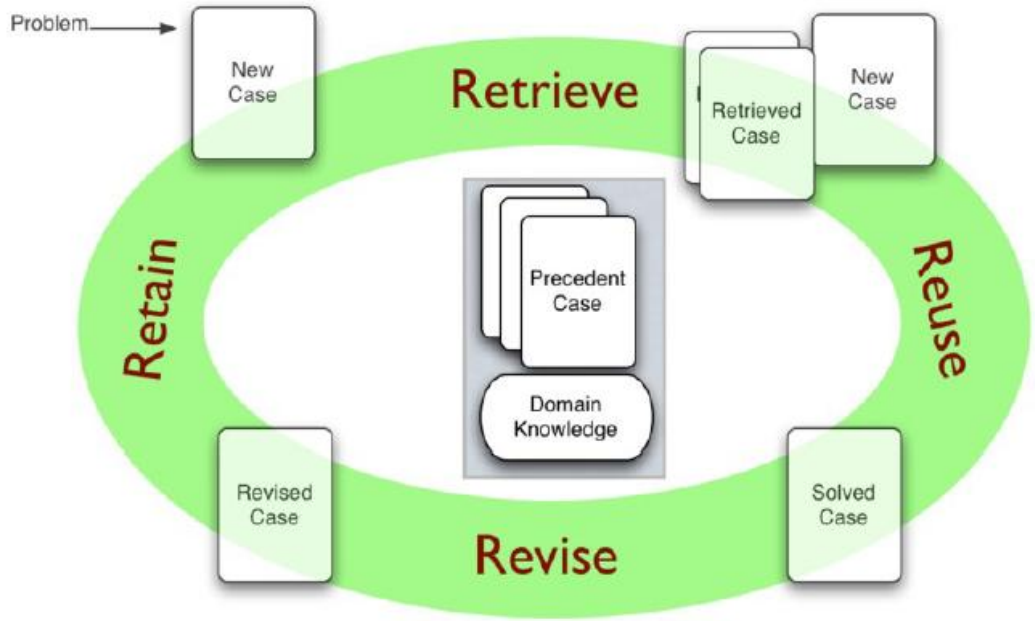
PLN se ha tenido que involucrar en el análisis semántico debido a las siguientes necesidades:

1. Traducción automática: originalmente, esta tarea se realizaba haciendo sustituciones directas entre palabras o frases, sin tomar en cuenta ningún criterio semántico. En los años 90 se contaban ya con recursos electrónicos que consideraban información léxico-semántica, p. e., el proyecto Verbmobil.



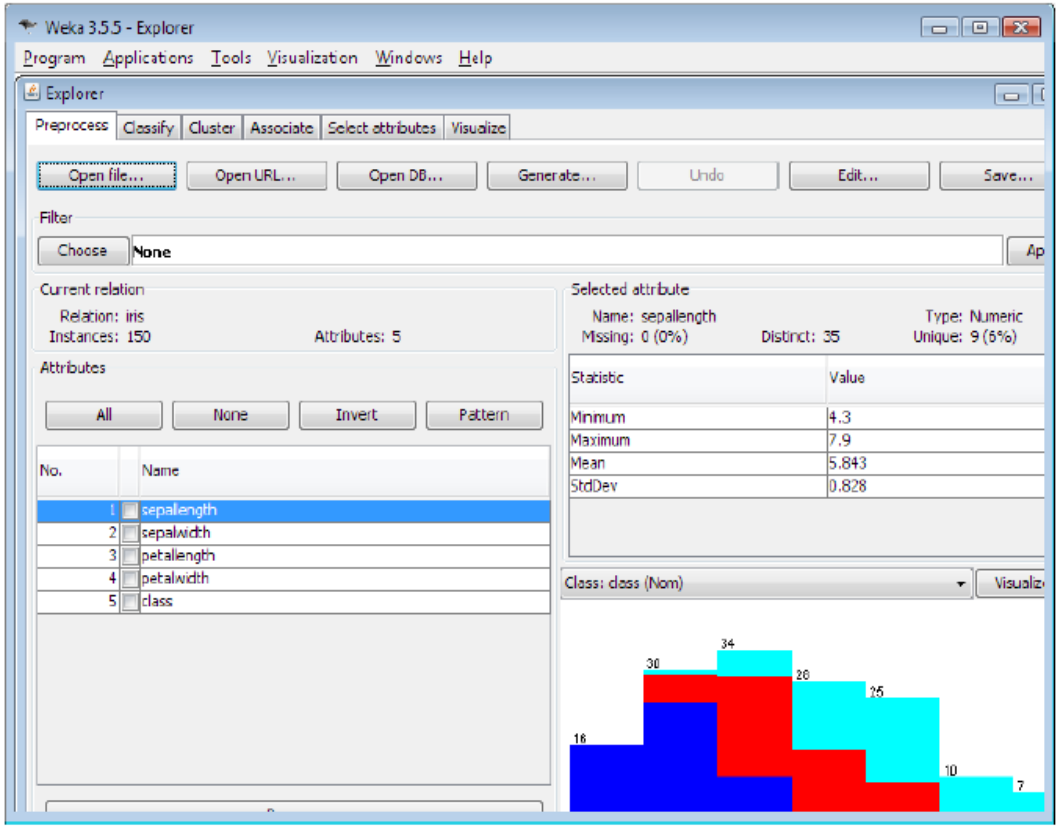
http://verbmobil.dfki.de/

¿Por qué queremos usar computadoras en cuestiones de semántica? (2)



2. Robótica y agentes inteligentes: debido a la complejidad que hay en la interacción entre máquinas y humanos, un camino que se ha explorado para mejorar el desempeño de las máquinas es brindarles información léxica y semántica para que concreten tareas específicas, vía el aprendizaje de conceptos.

¿Por qué queremos usar computadoras en cuestiones de semántica? (3)

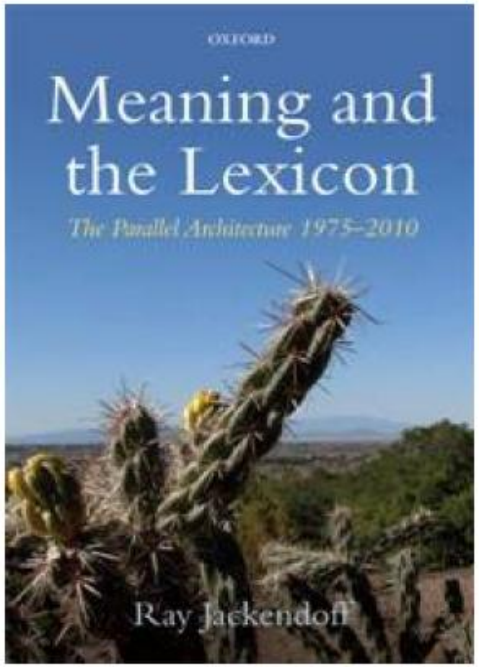


3. Búsqueda eficiente de información en Internet: dado el crecimiento exponencial de Internet, sabiendo además que es imposible regularlo, se ha pensado en sistemas de etiquetado y clasificación basados en el proceso que siguen los humanos para generar y comprender conceptos.

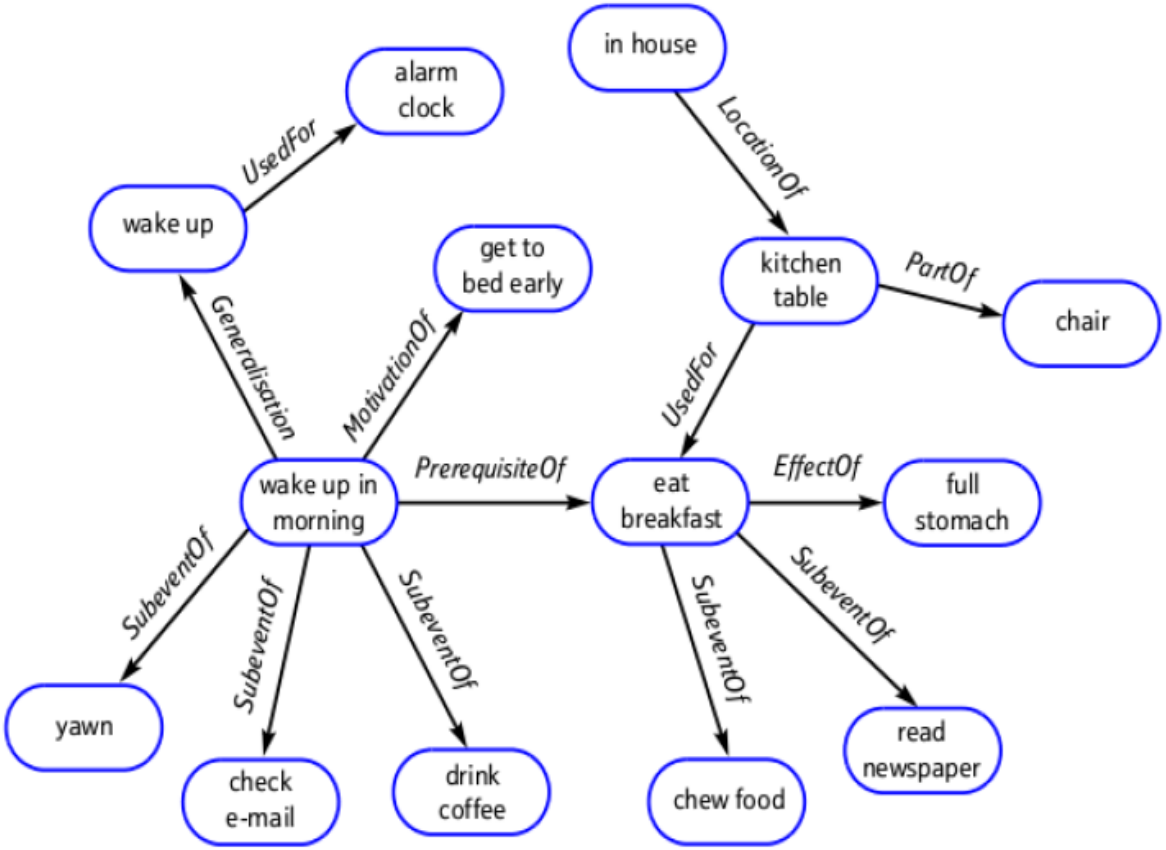
¿Por qué queremos usar computadoras en cuestiones de semántica? (4)

Este último punto ha tenido una gran relevancia en áreas tales como la extracción de información o la minería de textos, pues se parte de la base de que la información semántica contenida en un texto permite deducir descubrir nuevo conocimiento.

Tal idea ha sido desarrollada ampliamente por varios lingüistas. P. e., Ray Jackendoff (1948) delinea una arquitectura en donde el nivel semántico es la base en la cual se configuran conceptos.



Lengua = gramática + lexicón (1)



La semántica léxica parte de la idea de que, en nuestra mente, existe un súper-base de palabras, las cuales se combinan unas con otras para generar varias relaciones de sentido.

Estas combinaciones son controladas a partir de reglas, las cuales pueden ser tratables computacionalmente.

Lengua = gramática + lexicón (2)

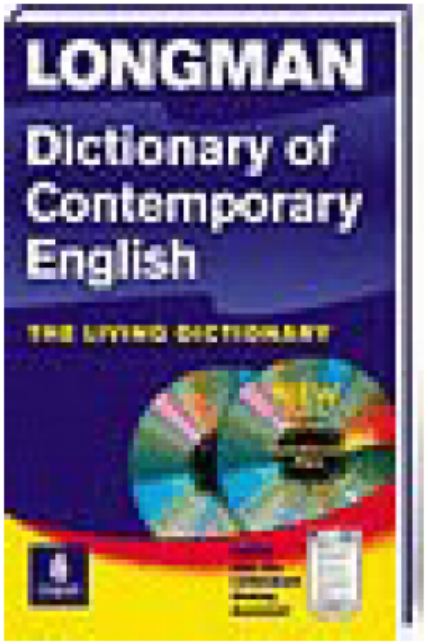
La semántica léxica computacional parte de la idea de reconocer y clasificar información conceptual a partir de relaciones que establecen relaciones jerárquicas que se dan entre palabras. Por lo general, se reconocer las siguientes:

Relation	Also called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	*breakfast*^{1} → *meal*^{1}
Hyponym	Subordinate	From concepts to subtypes	*meal*^{1} → *lunch*^{1}
Member Meronym	Has-Member	From groups to their members	*faculty*^{2} → *professor*^{1}
Has-Instance		From concepts to instances of the concept	*composer*^{1} → *Bach*^{1}
Instance		From instances to their concepts	*Austen*^{1} → *author*^{1}
Member Holonym	Member-Of	From members to their groups	*copilot*^{1} → *crew*^{1}
Part Meronym	Has-Part	From wholes to parts	*table*^{2} → *leg*^{3}
Part Holonym	Part-Of	From parts to wholes	*course*^{7} → *meal*^{1}
Antonym		Opposites	*leader*^{1} → *follower*^{1}

Lengua = gramática + lexicón (3)

Este conjunto de relaciones básicas han sido desarrolladas durante muchos años dentro del seno de la lexicografía. Por ello, una de las primeras áreas de aplicación de la semántica léxica ha sido la creación de diccionarios electrónicos.

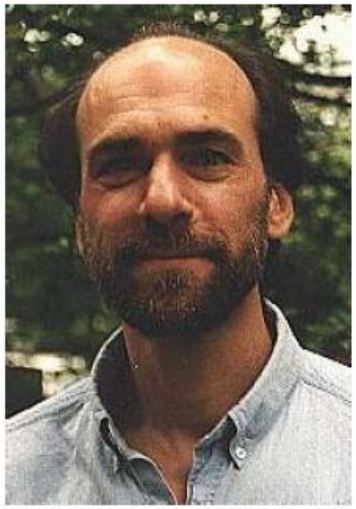
- Holonimia
- Polisemia
- Sinonimia
- Antonimia
- Hipernimia
- Hiponimia
- Meronomia

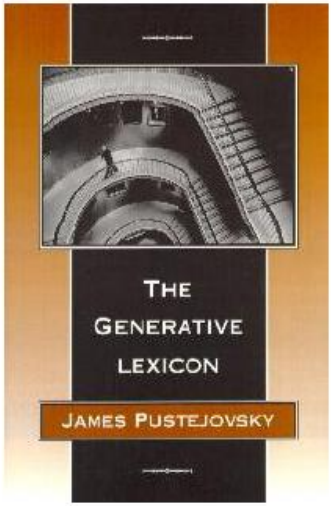


Un lexicón generativo (1)

Fue justo dentro del área de la lexicografía computacional que se ha desarrollado un modelo de representación léxica que ha sido muy explotado después. Este modelo se conoce como *lexicón generativo* (LG).

Aristóteles
(384-322 A.C.)





LG fue propuesto por James Pustejovsky, pensando como una vía para procesar información léxica de forma computacional. Lo interesante del modelo es que se basa en algunas ideas de Aristóteles.

Un lexicón generativo (2)

¿Qué retoma Pustejovsky de Aristóteles? Básicamente dos ideas:

Todo conocmiento puede ser expresado a partir de predicaciones, las cuales se derivan al seguir un pensamiento deductivo, inferido por medio de demostraciones.

1. La primera idea supone que todo conocimiento del mundo, al expresarse en lengua natural, se codifica siguiendo una estructura.

Premisas

1. Todos los hombres son mortales
1. Sócrates es un hombre

Conclusión/ demostración

1. Sócrates es mortal

Un lexicón generativo (3)

Categoría	Interpretación de la Categoría	Ejemplos
Esencia	*"¿Qué es X?"*	Hombre Sócrates
Cantidad	*"¿Cuánto es X?"*	4 metros 2 kilos
Calidad (Qualia)	*"¿Qué rasgos tiene X?"*	Blanco Soluble
Relación	*"¿Con qué se liga X?"*	Superior Inferior
Locación	*"¿Dónde está X?"*	En la escuela En el mercado
Tiempo	*"¿Cuándo es X?*	Ayer Hoy
Posición	*"¿Cómo está situado X?"*	Horizontal Paralelo
Posesión	*"¿Qué es propiedad de X?"*	Tiene patas Está armado
Acción	*"¿Qué hace X?"*	Corta Quema
Pasión (Pasivo)	*X es afectado por…*	Cortado Quemado

2. La siguiente idea es considerar que dichas estructuras se organizan dependiendo del tipo de información que expresan. Esto es, la famosa *teoría de las categorías*:

Un lexicón generativo (4)

LG, a grandes rasgos, puede ser descrito como una base de matrices léxicas asociadas a palabras, las cuales codifican información léxica conforme 4 niveles o estructuras:

1. **Argument Structure:** The behavior of a word as a function, with its arity specified. This is the predicate argument structure for a word, which indicates how it maps to syntactic expressions.
2. **Event Structure:** Identification of the particular event type (in the sense of Vendler [1967]) for a word or phrase: e.g. as state, process, or transition.
3. **Qualia Structure:** The essential attributes of an object as defined by the lexical item.
4. **Inheritance Structure:** How the word is globally related to other concepts in the lexicon.

Un lexicón generativo (5)

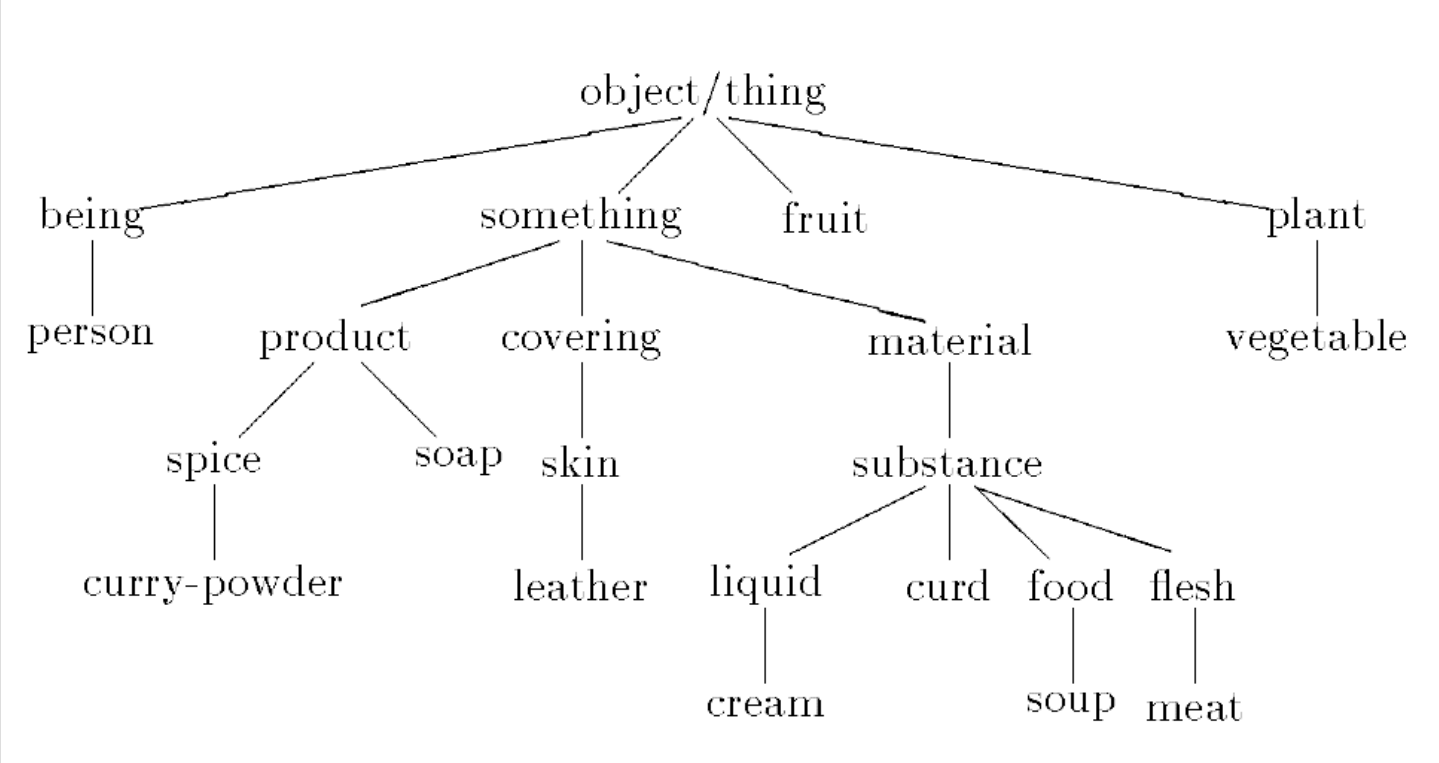
¿Cómo se relacionan estas 4 estructuras? En concreto, lo que queremos es establecer diferencias respecto al contenido léxico que puede asumir una palabra en dos o más contextos, p. e.:

a. John **baked** the potato.
$(bake_1 = change(x, State(y)))$
b. John **baked** the cake.
$(bake_2 = create(x, y))$

a. Mary **ran** yesterday.
$(run_1 = move(x))$
b. Mary **ran** to the store yesterday.
$(run_2 = go\text{-}to(x, y))$

Un lexicón generativo (6)

Cuando tomamos en cuenta esta clase de información, podemos pensar en la generación de taxonomías que nos ayuden a clasificar los conceptos que son referidos por las palabras de una lengua natural.



Ahora bien, hay que tener en cuenta que estas taxonomías **no son fijas, sino dinámicas**: son representaciones concretas de dominios de conocimiento que pueden ser generales y/o particulares.

Un lexicón generativo (7)

GL supone que toda esta información se codifica en matrices léxicas, las cuales (siguiendo a Aristóteles), describen los rasgos *necesarios y suficientes* para describir una palabra. Una matriz léxica se construye del siguiente modo:

$$\begin{bmatrix} \alpha \\ \text{ARGSTR} = \begin{bmatrix} \text{ARG1} = x \\ \ldots \end{bmatrix} \\ \text{EVENTSTR} = \begin{bmatrix} \text{E1} = e_1 \\ \ldots \end{bmatrix} \\ \text{QUALIA} = \begin{bmatrix} \text{CONST} = \textbf{what } x \textbf{ is made of} \\ \text{FORMAL} = \textbf{what } x \textbf{ is} \\ \text{TELIC} = \textbf{function of } x \\ \text{AGENTIVE} = \textbf{how } x \textbf{ came into being} \end{bmatrix} \end{bmatrix}$$

En este caso, la estructura qualia toma en cuenta 4 sub-rasgos: (1) Consistencia, (2) Forma, (3) Función y (4) Agentividad (esto es, cuál el origen o la génesis de nuestro concepto).

¿Cómo funcionan los rasgos Qualia? (1)

Veamos con mayor detalle cómo operan estos rasgos. Primero, delimitemos un objeto con ciertos rasgos, p. e., ser un objeto físico (*versus* un objeto abstracto):

$$\begin{bmatrix} \textbf{physobj(x)} \\ \text{FORMAL} = \textbf{physform(x)} \end{bmatrix}$$

Una vez hecho esto, consideremos que este objeto es fabricado por una persona, de modo que podemos considerarlo (de acuerdo con Pustejovsky) como un *artefacto*, el cual es capaz de desempeñar ciertas funciones (de acuerdo con el verbo al cual se asocie, p. e. *to eat* (representado como ***E*):**

$$\begin{bmatrix} \textbf{artifact_obj(x)} \\ \text{FORMAL} = \textbf{physform(x)} \\ \text{TELIC} = \textbf{Pred(E,y,x)} \end{bmatrix}$$

¿Cómo funcionan los rasgos Qualia? (2)

Finalmente, la matriz de rasgos nos permite representar cualquier artefacto que pueda ser consumible (dado que se asocia con la acción de comer); *ergo*, si seleccionamos la palabra *sandwich,* su matriz léxica sería:

$$\begin{bmatrix} \textbf{sandwich(x)} \\ \text{CONST} = \{\textbf{bread,...}\} \\ \text{FORMAL} = \textbf{physobj(x)} \\ \text{TELIC} = \textbf{eat(P,w,x)} \\ \text{AGENTIVE} = \textbf{artifact(x)} \end{bmatrix}$$

Observación: sin duda, estos no son los únicos rasgos que podemos considerar para *sandwich*, pero digamos que responden a preguntas básicas como: ¿qué es?, ¿de qué está hecho?, ¿qué forma asume?, ¿cómo funciona? , y finalmente ¿cuál es la naturaleza de su origen? Así, podemos considerar que esta matriz de rasgos es el equivalente formal a una entrada de diccionario.

¿Cómo construir una matriz léxica? (1)

Ahora, veamos cómo construir una plantilla léxica. Tomemos como ejemplo un verbo. La información que vamos a considerar de éste es:

1. Properties of the participants
2. Change (of being, state, location, relation)
3. Causation and means
4. Manner of the activity
5. Temporal and Locational constraints
6. Intentionality of the actor
7. Instrumental and co-agentive information
8. Psychological state of the participants

¿Cómo construir una matriz léxica? (2)

Una vez que hemos considerado la información anterior, ahora pasemos a analizar los componentes léxicos que proyecta un verbo:

Verb: V
Arguments: x, y, z, ...

Si queremos organizar esta información, respondamos a la pregunta:

Verb: V How do we decompose the meaning?
Arguments: x, y, z, ...

¿Cómo construir una matriz léxica? (3)

Resolver esta pregunta nos permite dividir los componentes del significado de un verbo en dos partes:

1. **Cuerpo**: la estructura del predicado, junto con todas sus variables ligadas.

2. **Entorno ligado:** la lista completa de parámetros contextuales en donde opera el verbo con sus argumentos.

a. Verb: V How do we decompose the meaning?
b. Arguments: x, y, z, ...

a. Body: the predicate, with bound variables.
b. Binding Environment (Args): the parameter list.

¿Cómo construir una matriz léxica? (4)

Ahora bien, establezcamos una estructura argumental básica, p. e.:

$$\lambda x_n \ldots \lambda x_1 [\Phi]$$

Esta fórmula lo único que describe es que un conjunto de argumentos X (desde X_0 hasta X_∞), constituyen un Predicado Φ.

En otras palabras, lo que obtenemos es:

$$\mathsf{Verb}(\mathsf{Arg}_1, \ldots, \mathsf{Arg}_n)$$

¿Cómo construir una matriz léxica? (5)

¿Qué análisis podemos realizar con las fórmulas anteriores? Veamos algunos ejemplos:

$$\lambda x[\mathsf{die}(x)]$$

The flower died.

$$\lambda y \lambda x[\mathsf{hit}(x, y)]$$

The car hit the wall.

¿Cómo construir una matriz léxica? (6)

Ahora bien, nosotros podemos añadir a nuestro predicado básico nuevos argumentos, según puedan ser requeridos:

$$\mathsf{Verb}(\mathsf{Arg}_1, \ldots, \mathsf{Arg}_n) \quad \Longrightarrow$$

$$\lambda x_m \ldots \lambda x_{n+1} \lambda x_n \ldots \lambda x_1 [\Phi]$$

Algunos ejemplos:

$\lambda y \lambda x \lambda e[\mathsf{kill}(e, x, y)]$: (Davidson, 1967)
The gardner killed the flower.

$\lambda z \lambda y \lambda x \lambda e[\mathsf{go}(e, x, y, z)]$: (Hobbs, 1993)
Nicholas went to China.

¿Cómo construir una matriz léxica? (7)

El símbolo Φ, que representa **Predicado**, define también expresiones complejas conformadas por sub-predicados fijados a un parámetro determinado:

$$\mathsf{Verb}(\mathsf{Arg}_1) \implies$$

$$\lambda x[\Phi_1, \ldots \Phi_k]$$

Este formalismo podemos aplicarlo tanto a verbos como a nombres, p.e.:

(9) a. **die**: $\lambda x[\mathsf{alive}(x) \wedge \mathsf{Become}(\neg\mathsf{alive}(x))]$

b. The flower died.

(10) **bachelor**:

$\lambda x[\mathsf{male}(x) \wedge \mathsf{person}(x) \wedge \mathsf{adult}(x) \wedge \neg\mathsf{married}(x)]$

¿Cómo construir una matriz léxica (8)

De acuerdo con Pustejovsky, este formalismo nos permite describir todos los predicados que pueden ser proyectados por una palabra, de tal suerte que esto nos ayuda a describir estructuras complejos a partir de sub-predicados:

$$\mathsf{Verb}(\mathsf{Arg}_1, \ldots, \mathsf{Arg}_n) \implies$$

$$\lambda x_m \ldots \lambda x_{n+1} \lambda x_n \ldots \lambda x_1 [\Phi_1, \ldots \Phi_k]$$

(12) a. **kill**:

$$\lambda y, x, e_1, e_2 [\mathsf{act}(e_1, x, y) \wedge \neg \mathsf{dead}(e_1, y) \wedge \mathsf{dead}(e_2, x) \wedge e_1 < e_2]:$$

De acuerdo con el ejemplo, *to kill* es un verbo que proyecta dos entidades, una que tiene el atributo de No-Muerta, y la segunda de Muerta, de tal modo que la primera afecta a la segunda.

Utilizando estructuras *Qualia* (1)

Después de haber ubicado los predicados potenciales asociados a una palabra, pasamos a describirlos en términos de estructuras *qualia*. Para ello, tenemos que considerar:

- La información semántica requerida para construir una matriz es regulada principalmente por dos estructuras: argumental y de eventos.
- Una vez que se delimitan estas dos estructuras, se pueden generar las demás estructuras qualia.
- Las reglas de composicionalidad suponen: selección, agrupamiento y coerción de tipos o clases.

Utilizando estructuras *Qualia* (2)

Una vez establecidas las condiciones anteriores, ¿cómo se relacionan las estructuras argumentales con las *qualia*?. Veamos:

$$\overbrace{\overbrace{\lambda x_n \ldots \lambda x_1}^{AS}\ \overbrace{\lambda e_m \ldots \lambda e_1}^{ES}}^{Environ}\qquad \overbrace{[Q_1 \wedge Q_2 \wedge Q_3 \wedge Q_4;\ C]}^{Body}$$

AS: Argument Structure

ES: Event Structure

Q_i: Qualia Structure

C: Constraints

Utilizando estructuras *Qualia* (3)

Recordemos en qué consisten las estructuras *qualia*:

Formal: the basic category which distinguishes it within a larger domain;

Constitutive: the relation between an object and its constituent parts;

Telic: its purpose and function;

Agentive: factors involved in its origin or "bringing it about".

Utilizando estructuras *Qualia* (4)

Cada una de estas estructuras ayuda a configurar una matriz del tipo:

$$\begin{bmatrix} \alpha \\ \text{ARGSTR} = \begin{bmatrix} \text{ARG1} = x \\ \ldots \end{bmatrix} \\ \text{EVENTSTR} = \begin{bmatrix} \text{EVENT1} = e1 \\ \text{EVENT2} = e2 \end{bmatrix} \\ \text{QUALIA} = \begin{bmatrix} \text{CONST} = \text{what } x \text{ is made of} \\ \text{FORMAL} = \text{what } x \text{ is} \\ \text{TELIC} = e_2\text{: function of } x \\ \text{AGENTIVE} = e_1\text{: how } x \text{ came into being} \end{bmatrix} \end{bmatrix}$$

Utilizando estructuras *Qualia* (5)

A partir de las estructuras *qualia* podemos reconocer 3 tipos de expresiones en el lenguaje natural:

- Natural Types: atomic concepts of FORMAL and CONST;
- Functional Types: Adds concepts of AGENTIVE and of TELIC;
- Complex Types: Cartesian types formed from both Natural and Functional types.

Nota: los tipos complejos son el resultado de deducir un producto cartesiano:

El **producto cartesiano** de A y B es el conjunto $A \times B$ cuyos elementos son los pares ordenados (a, b), donde a es un elemento de A y b un elemento de B:

$$A \times B = \{(a, b) : a \in A \text{ y } b \in B\}$$

Selección de rasgos (1)

Veamos algunos ejemplos sobre cómo se seleccionan rasgos para construir una frase u oración. P. e., supongamos que tenemos un conjunto de adjetivos, de los cuales señalamos algunas de sus propiedades:

Adjective	Qualia Selection
well-built	Agentive
unbaked	Agentive
red	Formal
stone	Constitutive
wooden	Constitutive
useful	Telic
carved	Agentive
effective	Telic
fast	Telic
heavy	Formal
dense	Const
large	Formal

Selección de rasgos (2)

Dependiendo del rasgo o atributo que queremos asociar a un nombre (p.e., *house*), generamos un tipo particular entidad:

(31)a. a well-built (A_1) house ($[F, C, A_1, T]$)

b. a two-story (F_1) house ($[F_1, C, A, T]$)

c. a vacation (T_1) house ($[F, C, A, T_1]$)

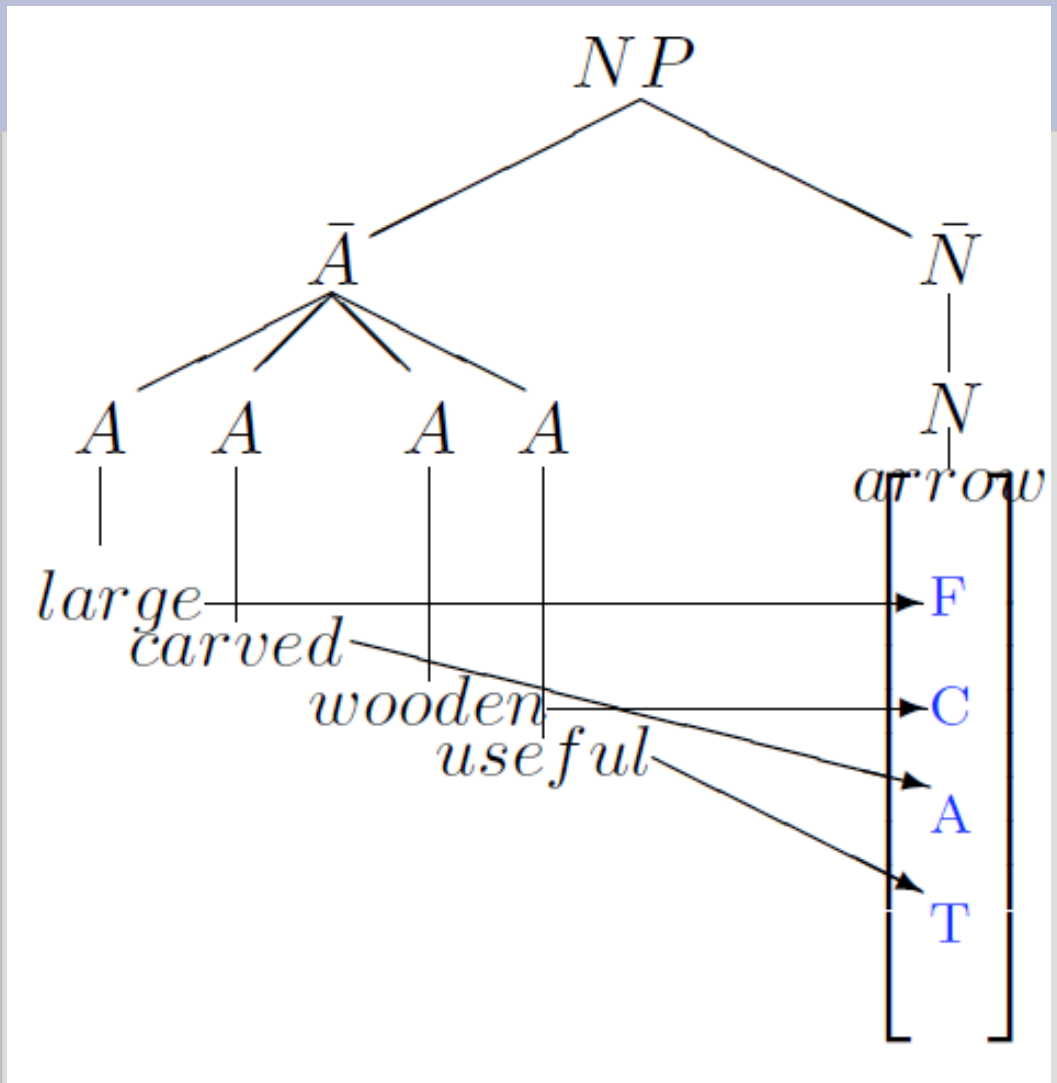
d. a brick (C_1) house ($[F, C_1, A, T]$)

(32)a. a large carved wooden useful arrow

b. a large (F_1) carved (A_2) wooden (C_3) useful (T_4) arrow ($[F_1, C_3, A_2, T_4]$)

Las letras *F*, *C*, *A* y *T* representan las estructuras *qualia* de *Forma, Constitución*, *Agentividad* y *Télica*.

Selección de rasgos (3)



Siguiendo en esto a Richard Montague y otros, cuando una palabra se inserta en el contexto de una estructura sintáctica, ésta selecciona a través de un proceso de mapeo qué rasgos son los requeridos para que dicha estructura adquiera su contenido semántico:

Selección de rasgos (4)

Considerando que se trata de un modelo de **generación** y **clasificación de significados**, Pustejovsky pondera el uso de formalismos lógicos que le permitan representar un nivel conceptual (en el sentido que lo entiende Jackendoff). P. e., una oración como *John broke a glass*, supone una representación lógica del tipo:

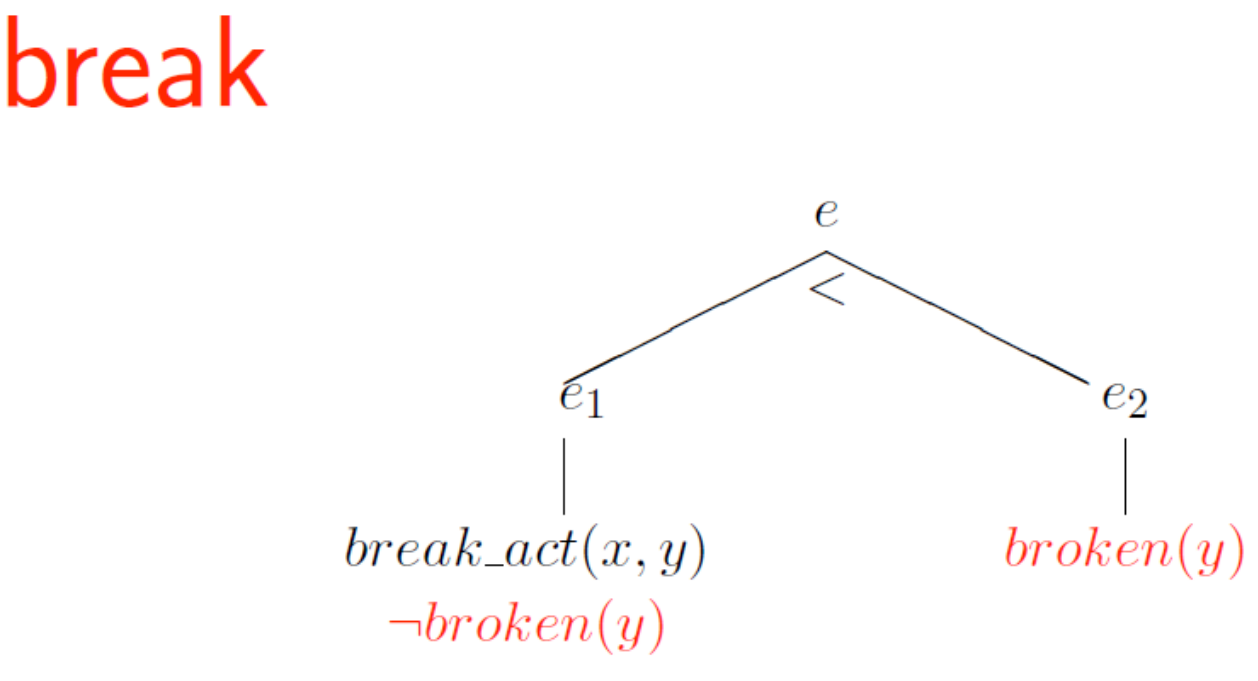
$$\exists x \exists e_1 \exists e_2 [glass(x) \wedge \neg broken(e_1, x) \wedge broken(e_2, x) \wedge e_1 < e_2]$$

Otra forma de representar esto es:

$$\overbrace{\exists x \overbrace{break_act}^{glass(x)} \quad \overbrace{broken}^{\neg glass(x)}}^{John}$$

Selección de rasgos (5)

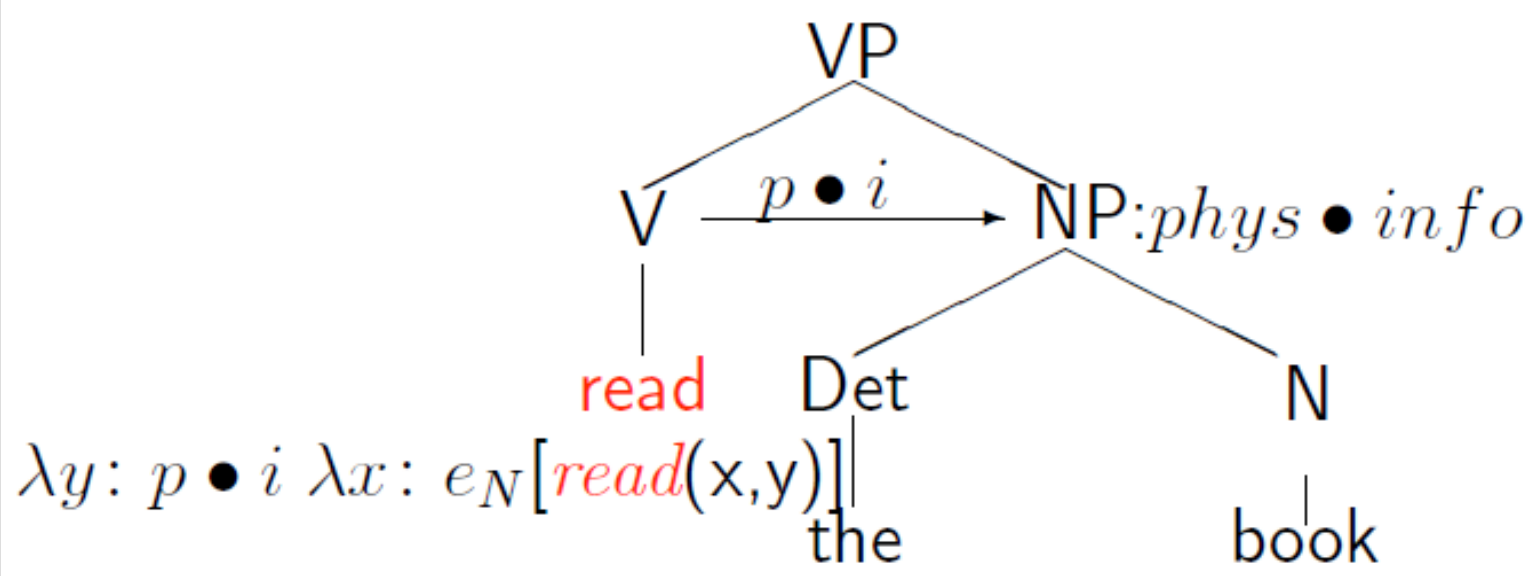
Dada esta selección de rasgos, podemos considerar que *to break* es un verbo que proyecta un cambio de estado, de acuerdo con este esquema:



Pregunta: si tomamos en cuenta este ejemplo, ¿qué pasa con construcciones como *coffe-break, lunch-break*, *cigarrete-break*, *tie-break* o *break-time*? En este sentido, LG nos ayudaría a analizar el significado de frases hechas, e incluso lexicalizadas.

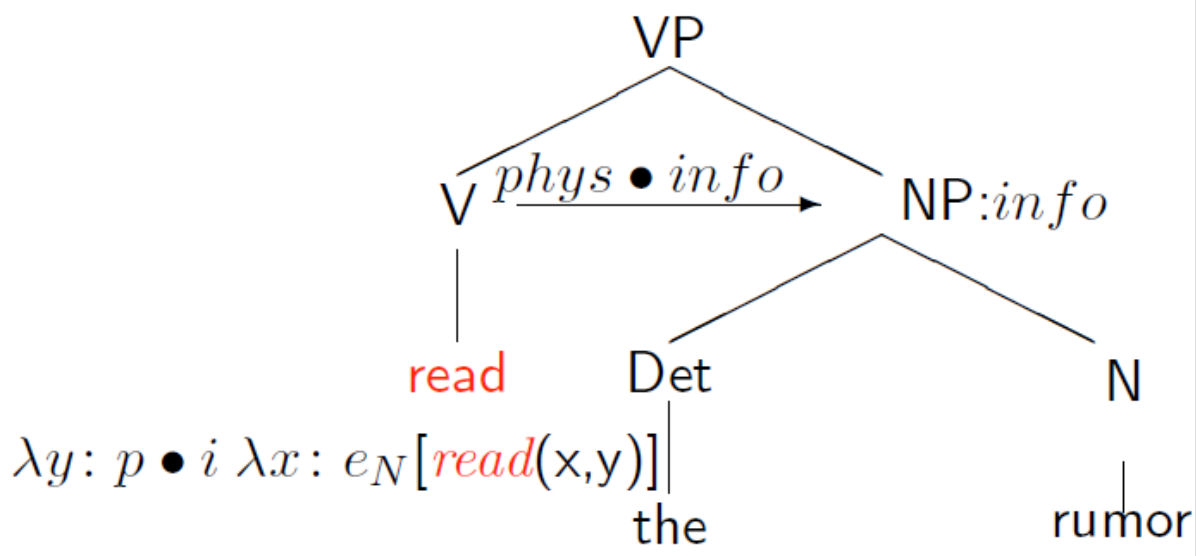
Selección de rasgos (6)

Ahora, como hemos señalado antes, LG considera que es posible generar una estructura sintáctica desde el mismo contenido léxico. P. e., el verbo *to read* supone una serie de rasgos que debe asumir la entidad que opere como argumento dentro del predicado de dicho verbo, esto es:



Selección de rasgos (7)

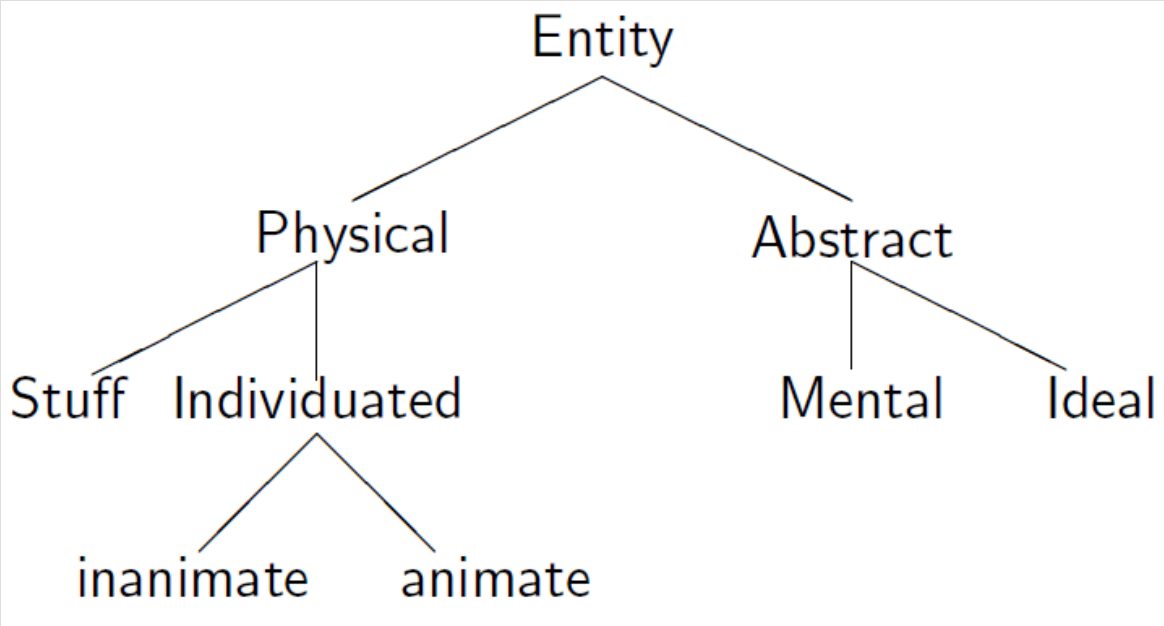
¿Qué pasa cuando el argumento selecciona otro ítem que no corresponde con el comportamiento esperado? Digamos que la selección no es caprichosa: por lo menos alguno de los rasgos considerados de la matriz léxica debe ser activado para establecer la relación, si no, tenemos un error de selección:



Comentario: en el caso de las colocaciones y las frases hechas, es interesante ver cómo LG propone un mecanismo de análisis alterno al que plantean modelos como la semántica cognitiva, en el sentido en que es posible explicar procesos metafóricos, considerando formalismos lógicos como los que emplea Pustejovsky.

Selección de rasgos (8)

Finalmente, la selección de rasgos nos proporciona información para desarrollar una clasificación de entidades y de eventos, tomando en cuenta las estructuras *qualia*, e igualmente el contexto sintáctico en el que se inserta una palabra:



Ahora bien, si se dan cuenta, estas taxonomías son al final **dinámicas**, pues las vamos generando **conforme a las estructuras *qualia* que se activan (o desactivan) en una estructura sintáctica dada**.

Representación de eventos (1)

Dado cierto contexto, podemos escoger los rasgos que sean acordes para construir un evento. Por ejemplo, supongamos que tenemos la siguiente entrada léxica:

```
door(*x*,*y*)
    Const: aperture(*y*)
    Form: phys-obj(*x*)
    Telic: pass-through(T,z,*y*)
    Agentive: artifact(*x*)
```

Tomando en cuenta esta entrada para *door*, la podemos emplear para construir 3 eventos distintos:

The door is **closed**.
The door **closed**.
John **closed** the door.

Representación de eventos (2)

Cada una de estas oraciones representa un tipo de evento distinto, en los cuales *door* asume estos rasgos. P. e., cuando pensamos en la primera oración, lo que hacemos es describir un estado, esto es:

State (S): a single event, which is evaluated relative to no other event
Examples: *be sick, love, know*
Structural representation:.

$$\begin{array}{c} S \\ | \\ e \end{array}$$

En cambio, la segunda es un proceso, descrito de la siguiente forma:

Process (P): a sequence of events identifying the same semantic expression
Examples: *run, push, drag*
Structural representation:

$$\begin{array}{c} P \\ \triangle \\ e_1 \ldots e_n \end{array}$$

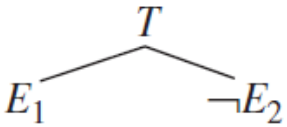
Representación de eventos (3)

Finalmente, tenemos eventos que se suscriben a una transición temporal (p.e., generar un cambio de estado, o aplicar un proceso para generar o modificar una entidad). El esquema que los representa es:

Transition (T): an event identifying a semantic expression, which is evaluated relative to its opposition (Jackendoff, 1972; Lakoff, 1970; von Wright, 1963)[12]

Examples: *give, open, build, destroy*

Structural representation (where E is a variable for any event type):



Nota: siguiendo el esquema de clasificación de eventos propuesto por Zeno Vendler (1921–2004), Pustejovsky plantea que los *accomplishments* (logros) y *achievements* (realizaciones) pueden integrarse en una sola categoría, justo la de las transiciones.

Representación de eventos (4)

Considerando los tres esquemas anteriores, podemos asignar uno en particular. En el caso la oración *the door is closed* tenemos:

Las abreviaturas que empleamos aquí son:

1. **ES** = Event Structure
2. **LCS** = Lexical-Conceptual Structure

The door is closed.

ES: S

LCS′: e

LCS: [closed(the-door)]

[closed(the-door)]

Pregunta 1: tomando en cuenta la estructura de este evento, ¿los rasgos léxicos asociados a *door* se conservan en su totalidad, o se prioriza algún rasgo particular?.

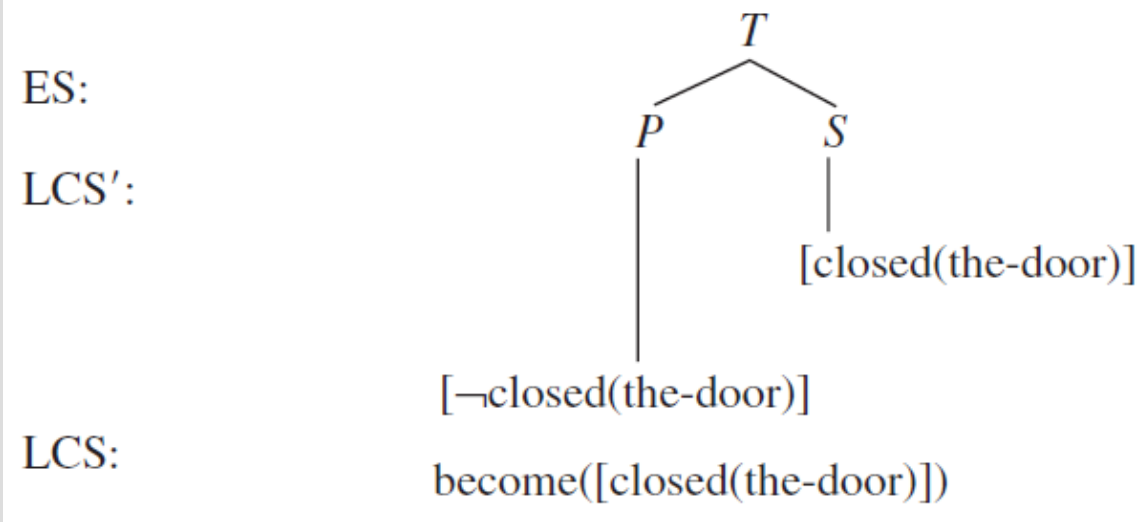
Representación de eventos (5)

Ahora, en el caso de oración *the door closed* (en español, *la puerta se cerró*, ya que requiere la inserción del clítico *se*), el esquema es:

En este esquema, *T* representa una transición, *P* es un proceso y *S* es un estado. Así, lo que expresa el esquema es que el proceso de **cerrar la puerta** supone cambio de estado (p.e., la puerta tiene el atributo de **no-cerrada**, y luego asume el atributo **cerrada**).

Nuestra LCS es la que aparece al final del esquema.

The door closed.



Pregunta 2: ¿*Door* conserva sus rasgos léxicos, o hay alguna clase de selección?

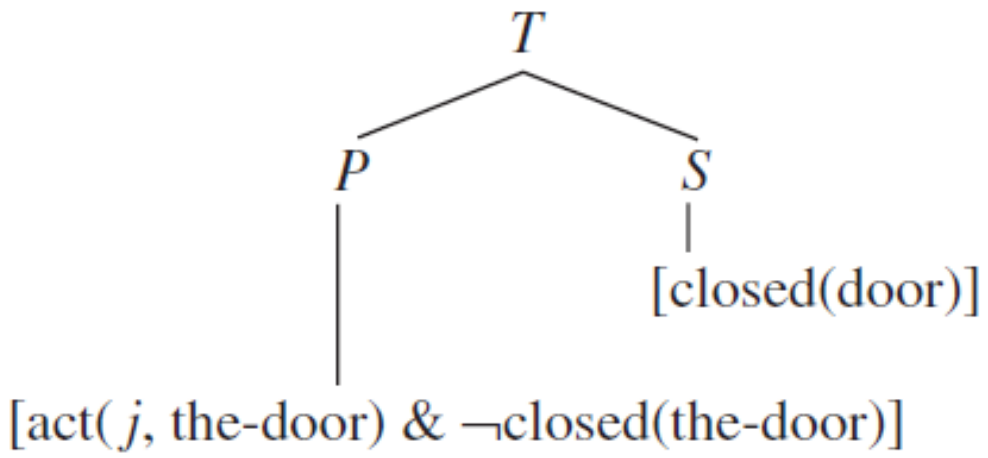
Representación de eventos (6)

Finalmente, la oración *John closed the door* es vista como una transición plena, de modo que su esquema es:

John closed the door.

ES:

LCS′:



LCS:

cause([act(j, the-door)], become([closed(the-door)]))

Pregunta 3: ¿*Door* conservó siempre sus rasgos léxicos?

Gracias por su atención

Blog del curso:
http://cesaraguilar.weebly.com/curso-de-procesamiento-del-lenguaje-natural.html